SYNTAX ZX80®

A PUBLICATION OF **THE HARVARD GROUP**

VOL.2 NO.7 ISSN 0273-2696 JUL., 1981

SINCLAIR TO REPLACE FAULTY 8K ROMS

Have you been waiting for your 8K ROM from Sinclair? It may be a few more weeks. A new ROM supplier produced 8K ROMs with a bug, and Sinclair has stopped shipping until a supply of perfect ROMs arrives.

If you own one of the faulty ROMs that slipped through, Sinclair will replace it for you. Nigel Searle of Sinclair explained, "Some 8K ROMs have a bug. We are waiting for a new batch of correct ROMs from our supplier. This may take several weeks. In the meantime, please write your name, address, and the words, "8K ROM" on a piece of paper and send it to Sinclair Research, 50 Staniford St., Boston, MA, 02114. We will then send you a new ROM as soon as they are available. We apologize for any inconvenience you may have been caused." Nigel emphasized that you should keep your present 8K ROM until it is replaced and the bug should cause no problems during normal programming. It seems to appear in some arithmetic calculations of very large numbers. See David Shulman's letter p.8 this issue.

SYNTAX ARTICLE REPRINTS

Now you can order specific article reprints from SYNTAX. We don't have a complete articles index yet, but if you see a story referenced and would like a copy, send us the title or subject and the month it appeared. Include your name, address, and a check for $1 (don't mail cash). It will be on its way first class the same day we receive your order.

FULL ANNOTATED 4K ROM LISTING

Sinclair's complete 4K ROM listing with original designer's annotations will be available from SYNTAX by August. Pre-publication price only $29. After publication, $40. Send check or credit card number (MasterCard, Visa, Amex, Diner's) with expiration date to SYNTAX, RD 2 Box 457, Bolton Rd., Harvard, MA, 01451 or call 617/456-3661. Shipped postpaid.

NEWS FROM MICROACE

MicroAce plans to sell plug-in 16K RAM boards with 3 extra connector slots. The extra connectors will be 50 pin, not 46 (as the rear edge connector is), because of the unavailability of 46 pin plugs. They will ship in August or September. The RAM boards will cost $149. Contact MicroAce at 1348 E. Edinger, Santa Ana, CA, 92705, 714/547-2526 for more information.

MicroAce is still without a full-time manager, but Bill Clark, engineer, is serving temporarily until they find a full-timer.

4K RAM UNITS AVAILABLE SOON

Static Memory Systems of Freeport, IL, will sell 4K RAM packs for ZX80s and MicroAces. According to Norman Wheelock of Static Memory, the white, 1.5" tall units will cost $49.95 and provide an additional buffered connector on the side. Static Memory expects to ship in August. Contact Static Memory Systems at 15 S. Van Buren, Suite 209, Freeport, IL, 61032, 815/235-8713 for more information.

MAZE OPTIONS

Otis Imboden's 2-line Maze-O-Graphics (Mar.81) inspired several readers. Here are their 4K ROM maze improvements, offering different methods. Dan O'Connell's adds shaded graphics for "trap doors." Try these as bases for games.

```
   5 LET X=1
  10 LET Z=RND(4)
  20 GO TO 10*Z+50
  60 PRINT "▄";              (shift W)
  65 GO TO 100
  70 IF NOT X=4 THEN PRINT "▌";
(shift Q)
  71 IF X=4 THEN GO TO 80
  75 GO TO 100
  80 IF (X=1 OR X=4) AND NOT (X=
2 OR X=3) AND NOT PEEK(16420)<2
THEN PRINT "▖";             (shift D)
  85 GO TO 100
  90 IF NOT X=4 THEN PRINT CHR$(
133);
  95 IF X=4 THEN GO TO 60
 100 LET X=Z
 105 GO TO 10
Syntactic Sum: 18666, 4K
```

J. Alexander Jr., Scotch Plains, NJ

```
   5 FOR J=1 TO 352
   6 LET A=RND(11)
  10 PRINT CHR$(A);
  20 PRINT CHR$(2);CHR$(131);
  30 NEXT J
Syntactic Sum: 5996, 4K
```

D.J. O'Connell Jr., San Antonio, TX

```
  10 DIM A(2)
  20 LET A(1)=RND(2)+1
  30 LET A(2)=(RND(2)-1)*128+5-A
(1)
  40 LET B=128+A(1)
  50 LET C=RND(25)
  60 IF C=1 THEN GO TO 90
  70 PRINT CHR$(A(RND(2)));
  80 GO TO 50
  90 PRINT CHR$(B);
 100 GO TO 50
Syntactic Sum: 15204, 4K
```

John P. Filley, Nashville, TN

SYNTAX ERROR: Biorythym, Jan.81, contained an error. Line 106 should read:

```
 106 LET P2=2*π
```

The Jan. version spelled out PI. Use the keyboard symbol instead.

ZX81, SINCLAIR PRINTER NEWS:

Sinclair has submitted the ZX81 to the FCC for approval for sale in the US. Nigel Searle of Sinclair expects to ship American ZX81s this year, maybe September.

Their printer, scheduled to appear in Europe in June, has not gone into production.

OUR POLICY ON CONTRIBUTED MATERIAL

SYNTAX ZX80 invites you to express opinions related to the ZX80 and the newsletter. We will print, as space allows, letters discussing items of general interest. Of course, we reserve the right to edit letters to a suitable length and to refuse publication of any material.

We welcome program listings for all levels of expertise. Programs can be for any fun or useful purpose. We will test run each one before publishing it, but we will not debug programs; please send only workable listings.

In return for your listing, we will pay you a token fee of $2.00 per program we use. This payment gives us the nonexclusive right to use that program in any form, world-wide. This means you can still use it, sell it, or give it away, and so can we.

We will consider submissions of news and hardware or software reviews. Please keep articles short (350-400 words). Again, we reserve the right to edit accepted articles to a suitable length. We will pay 7 cents per 6 characters, including spaces and punctuation, for accepted articles.

When you send in programs for possible publication in SYNTAX, please include the following information:

- How to operate the program, including what to input if it does not contain prompts.
- Whether you can run the program over again and how.
- How to exit the program.
- The Syntactic Sum (using the Syntactic Sum program in the February, 1981, issue).
- Whether it fits in 1K or 2K RAM (or 16K when available).
- Whether it uses the 4K or 8K RAM.

We pay for this explanatory text at the same rate as for articles in addition to payment for the program itself.

If you want us to return you original program listing or article, please include a self-addressed, stamped envelope. Otherwise, we cannot return submitted material.

SYNTAX ZX80 is published monthly by Syntax ZX80, Inc., a wholly-owned subsidiary of The Harvard Group, RD 2, Box 457, Bolton Road, Harvard, MA 01451. Telephone 617/456-3661. Subscriptions: 12 issues, $25. Single issue, $4.

Publisher: Kirtland H. Olson
Editor: Ann L. Zevnik
Associate Editor: Susan G. Barber
Technical consultant: Eric K. Olson

Printed by Joseph E. Marhefka, Jr.
Clinton Offset Printers
Clinton, MA 01510

WRITING GAME PROGRAMS

How do you start writing game programs? Surprisingly, the game often only develops while actually writing the program. This back-to-front method is not suited to larger machines, but it works well for the ZX80. Let's write a game.

I often start by drawing a board on the screen. Enter:

```
  30 GO SUB 140
  40 FOR A=1 TO 5
  50 PRINT CHR$(128);
  60 FOR B=1 TO 30
  70 PRINT CHR$(0);
  80 NEXT B
  90 PRINT CHR$(128)
 100 NEXT A
 110 GO SUB 140
 120 CLEAR
 130 GO TO 180
 140 FOR C=1 TO 32
 150 PRINT CHR$(128);
 160 NEXT C
 170 RETURN
RUN
```

The subroutine at 140 draws either the top or bottom lines. I space the lines well apart, in tens; use variables in order, A, B, C; and use the CLEAR instruction where variables are no longer needed. CLEAR reclaims RAM that might be useful later on.

Changing line 40 makes the board larger. Changing lines 60 and 140 make it squarer.

What are the addresses of the current board's squares? Square addresses in the middle line are given by:

```
 190 LET E=0
 210 INPUT F
 240 LET E=100+PEEK(16396)+PEEK(
16397)*256+F
 250 POKE E,50
```

Line 190 defines a variable E that will hold the value of the actual address. It will probably end up around 17000, as a guess. INPUT F allows the numbers of the squares in the middle row to be in the range 1 to 30. So enter only numbers between 1 & 30 in reply to the prompt when choosing a square in the middle line.

Line 240 is the standard way to find the address of a location in the display file. (E must be previously defined.)

The offset, 100 in this case, goes along the display file 100 locations.

Finally, add in the distance along the line, F.

Line 250 will put an "M" in the chosen square.

It's important to define no new variables between the address calculation (line 240) and any line in which it is used. If you do, the calculated addresses will be wrong.

Now, at last, the idea of this game is forming. Let the computer choose a square on the middle line and then let the player try to find it. A line like:

```
 180 LET D=RND(30)
```

will certainly do the choosing for a 4K ROM. And a line like:

```
 260 IF ABS(D-F)>2 THEN GO TO 21
0
```

allows the player to win if he gets either of the 2 squares before, the square itself, or the 2 after. (ABS makes any number positive.) So par for the game is a win in 6 tries.

Now we tidy it all up and put in some frills. First, a title. Something like "BOMBER OVERHEAD" might do--to play you try to shoot the invisible bomber out of the sky. Add these lines:

```
  10 PRINT "******  BOMBER OVER
HEAD  ******"
  20 PRINT
```

Now we must change the offset in line 240. No longer is it 100, but 134 because we added 34 extra characters to the display file before the board:

```
 240 LET E=134+PEEK(16396)+PEEK(
16397)*256+F
```

To add some instructions:

```
 200 PRINT "SHOOT 1-30"
```

(or something more elaborate if

you wish.)

The input should be validated so that you cannot enter numbers with no meaning to this program, so add:

```
 230 IF F<1 OR F>30 THEN GO TO 2
10
```

The computer will wait for a valid entry.

Another important feature to add erases your last shot:

```
 220 POKE E,0
```

Note that on the first run through 0 is poked into location 0 since E is initially set to 0. This can sometimes lead to problems. You can try to POKE into ROM addresses, 0-4095, because nothing will actually happen. But in some programs you might want to initialize E to some other value. For example, initializing E to 16384 (the first RAM address) would be asking for trouble. (Note--see PEEK-POKE article, Jun.81--AZ)

So far our program falls through the loop 210-260 if the target is hit, but goes round and round the loop if the shot misses. We need some lines to show a hit. Try, for example:

```
 270 POKE E,20      (the square)
 280 POKE E-33,20   (above)
 290 POKE E+33,20   (below)
```

We have 1 slight problem to overcome if the squares before and after are to be filled on success. I do not like to see the board being overwritten--it's bad programming. So test for the edge-condition (if the square is at the edge) before POKEing the "before" and "after" squares:

```
 300 IF NOT F=1 THEN POKE E-1,20
 310 IF NOT F=30 THEN POKE E+1,2
0
```

Let's add an end message:

```
 320 PRINT
 330 PRINT "**  A GONER  **"
```

and a restart mechanism:

```
 340 INPUT A$
 350 CLS
 360 RANDOMISE
 370 RUN
```

Syntactic Sum: -25280 4K

This program is very simple, but it shows good features. POKEing to the display file is easy as long as you don't break the rules.

Ian Logan, Skellingthorpe, UK

PUBLICATIONS GUIDE

The spring Westlake Guide lists over 350 computer, technology, electronics, and business magazines, newsletters, and report/database services. Westlake offers a trial plan in which you can sample listed publications at low cost (refundable if you subscribe). For a free copy, send your name and address to Westlake Subscription Service, 4200 S. Louise Ave., Sioux Falls, SD, 57106, or call 605/331-6930.

ANNOTATED 4K ROM: LOAD ROUTINE

This month we're listing an assembly for the 4K ROM's LOAD routine, next page. LOAD begins in ROM location 0206H, or 518d. As usual, we've left the designers' comments alone.

We initialized the system variable E-LINE (listed as E8LINE) at 400AH and subroutines ENDBYT and LS8, which occur between LOAD and SAVE (see May 81, pp. 5-6). Our computer won't assemble LOAD until we define all subroutines outside LOAD, so remember that none of the EQU statements listed are actually in the ROM.

LOAD depends on what SAVE has done, so if you've modified SAVE for your own use, you may have to change memory addresses or timing. If you modify LOAD, you'll also have to check SAVE.

To get the entire 4K ROM listing at once, see page 1.

```
0205           00100          ORG     0205H
0203           00110 LS8      EQU     0203H
4000           00115 Y        EQU     4000H
400A           00120 E8LINE   EQU     400AH
01F8           00130 ENDBYT   EQU     01F8H
0205 00        00140 LOAD     NOP             ;READ TAPE WRITTEN BY SAVE
0206 D1        00150          POP     DE
0207 111257    00160 LL1      LD      DE,22290
020A 3E7F      00170 LL2      LD      A,7FH   ;LOOK FOR LEAD-IN,THEN
020C DBFE      00180          IN      A,(0FEH)        ;READ EVERY 15.1 MSEC
020E 1F        00190          RRA             ;IF 1'S, 21.5: IF 0'S, GET 0'S
020F 30F2      00200          JR      NC,LS8  ;FOR 480 MSEC,TO NEWLINE IF
0211 17        00210          RLA             ;BREAK PRESSED
0212 17        00220          RLA
0213 38F2      00230          JR      C,LL1
0215 1B        00240          DEC     DE
0216 7A        00250          LD      A,D
0217 B3        00260          OR      E
0218 20F0      00270          JR      NZ,LL2
021A FD340B    00280          INC     (IY+E8LINE+1-Y) ;IN CASE IS 40H NOW
021D 210040    00290          LD      HL,4000H        ;NOW START READING DATA
0220 1E08      00300 LL27     LD      E,8
0222 3E7F      00310 LL3      LD      A,7FH
0224 DBFE      00320          IN      A,(0FEH)
0226 1F        00330          RRA
0227 3024      00340          JR      NC,LL7
0229 17        00350          RLA
022A 17        00360          RLA
022B 30F5      00370          JR      NC,LL3  ;READ EVERY 16 MSEC UNTIL '1'
022D 0E94      00380          LD      C,148
022F 061A      00390 LL4      LD      B,26    ;REQ 408 MSEC TRAILER AFTER TONE
0231 0D        00400 LL5      DEC     C       ;COUNT 15.4 MSEC IF 1, 15.2 IF 0
0232 DBFE      00410          IN      A,(0FEH)        ;NB C DECREMENTED AT
0234 17        00420          RLA             ;LEAST 26 TIMES
0235 CB79      00430          BIT     7,C
0237 79        00440          LD      A,C
0238 38F5      00450          JR      C,LL4
023A 10F5      00460          DJNZ    LL5     ;AT END OF TRAILER, CLEAR CARRY
023C 2004      00470          JR      NZ,LL6  ;IF > ABOUT 2.3 MSEC, IS A '1'
023E FE56      00480          CP      86
0240 30E0      00490          JR      NC,LL3  ;IF < ABOUT 0.8 MSEC, IS NOISE
0242 3F        00500 LL6      CCF             ;CARRY SET IF 0, CLEAR IF 1
0243 CB16      00510          RL      (HL)
0245 1D        00520          DEC     E
0246 20DA      00530          JR      NZ,LL3  ;GO GET NEXT BIT
0248 CDF801    00540          CALL    ENDBYT  ;AFTER 8 BITS
024B 18D3      00550          JR      LL27
024D 15        00560 LL7      DEC     D
024E F20000    00570          JP      P,0     ;IF HAVE BEEN THROUGH ENDBYT
0251 FD350B    00580          DEC     (IY+E8LINE+1-Y) ;ELSE RESTORE E8LINE
0254 18AD      00590          JR      LS8
```

LOAN AMORTIZATION SCHEDULE

This 8K/16K program gives you monthly payments and loan amortization information. To use, type in the program and hit RUN (NL). Answer the prompts, giving your yearly percentage rate as a decimal (thus enter 11% as .11).

The ZX80's 32-column display width limits the amount of data the schedule tables can hold. At high interest rates (a relative term) the display can't print the total interest column on 1 line.

Lines 65-75 calculate your monthly payments. The subroutine at 500 is a round off routine. Lines 140-240 provide the table of principal and interest payments:
135 Initialize for amort. sched.
150 Loop for number of payments
155 Compute interest
175 Compute principal
180 Force out last payment
200 Update loan balance
220 Update total interest
225 Allow table to print 1 line at a time. Increase number for slower display, delete line for full screen at once.
230 Print data line
235 Holds display after every 20 data lines until user hits NL, then fills new screen until M data lines (payments). Subroutine at 520 waits for NL to continue.

Mel Routt, Safety Harbor, FL

```
  1 REM "AMORTIZATION"
  2 REM BY MEL ROUTT, JUNE 81
  3 REM ADAPTED FROM PG 98
  4 REM "BASIC PRIMER"
  5 REM COPYRIGHT 1978
  6 REM BY MITCH WAITE
  7 REM AND MICHAEL PARDEE
  8 REM HOWARD W SAMS CO INC
  9 REM FIRST EDITION
 10 CLS
 15 PRINT " ENTER:"
 20 PRINT "AMOUNT OF LOAN: $";
 25 INPUT L
 30 PRINT L
 35 PRINT "ANNUAL O/O RATE: ";
 40 INPUT R
 45 PRINT R
 50 PRINT "TERM IN YEARS:   ";
 55 INPUT Y
 60 PRINT Y
 65 LET I=R/12
 70 LET M=Y*12
 75 LET A=L*(I*(1+I)**M)/((1+I)
**M-1)
 80 LET X=A
 85 GOSUB 500
 90 LET A=X
 95 PRINT "MONTHLY PAYMENT=
$"; A
 100 PRINT
 105 PRINT "IF YOU WISH TO VIEW
PAYMENT    SCHEDULE, TYPE ""YES
"". A ""NO""   WILL PERMIT A NE
W ENTRY."
 110 INPUT A$
 115 IF A$="NO" THEN GOTO 125
 120 CLS
 130 PRINT " SCHEDULE:"
 135 GOSUB 510
 140 LET B=L
 145 LET T=0
 150 FOR J=1 TO M
 155 LET I1=B*I
 160 LET X=I1
 165 GOSUB 500
 170 LET I1=X
 175 LET P=A-I1
 180 IF J=M THEN LET P=B
 200 LET B=B-P
 205 LET X=B
 210 GOSUB 500
 215 LET B=X
 220 LET T=T+I1
 225 PAUSE 20
 227 POKE 16437,255
 230 PRINT J;TAB 3;I1;TAB 10;P;T
AB 16;B;TAB 25;T
 235 IF J=60 OR J=40 OR J=20 THE
N GOSUB 520
 240 NEXT J
 245 PRINT
 250 PRINT "WANT TO PROCESS ANOT
HER LOAN?   ""YES"" OR ""NO"""
 255 INPUT A$
 260 IF A$="YES" THEN GOTO 10
 270 CLS
 275 PRINT
 280 PRINT "IT WAS A PLEASURE TO
 SERVE YOU."
 285 STOP
 500 LET X=(INT (100*(X+(.005)))
)/100
 505 RETURN
 510 PRINT "NO:";TAB 4;"INT:";TA
B 10;"PRIN:";TAB 16;"P.BAL:";TAB
 25;"T.INT:"
 515 RETURN
 520 INPUT X$
 525 IF X$="" THEN CLS
 530 GOSUB 510
 535 RETURN
```

Syntactic Sum: 8420 8K

ZX80 CHECK BOOK

This program uses 4K ROM & 1K RAM to record up to 10 transactions at a time (8K modifications below). Keep them on file by storing the program with the variables intact. Just SAVE the program after using it, then type GO TO 10, not RUN, to run after reloading.

RUN the program and follow the prompts. You can number deposits with a flag like 9999 or a code for the date. Days with 1 digit should have a zero in front of them. The register can be checked after each entry by hitting 0 (NL). To continue adding entries hit NL. To exit, type STOP.

Drop the decimal point on all numbers and add them back when transferring transactions to your checkbook. Do not drop zeros; enter $1.00 as 100.

Timothy Sorber, FPO New York

```
  10 DIM A(9)
  20 DIM B(9)
  30 FOR L=0 TO 9
  40 LET A(L)=0
  50 LET B(L)=0
  60 NEXT L
  70 PRINT "PREV BALANCE?"
  80 INPUT B
  90 FOR K=0 TO 9
 100 PRINT "REGISTER(0), DEPOSIT
(1)"
 105 PRINT "OR CHECK(2)?"
 110 INPUT T
 120 IF T=0 THEN GO TO 220
 130 PRINT "NUMBER?"
 140 INPUT N
 150 LET D1=T*10000+N
 160 LET A(K)=D1
 170 PRINT "AMOUNT?"
 180 INPUT D2
 190 LET B(K)=D2
 200 CLS
 210 NEXT K
 220 CLS
 230 PRINT "   ZX80 CHECK BOOK"
 240 PRINT
 250 PRINT "PREV BALANCE",B
 255 PRINT
 260 PRINT "NR","AMOUNT","BALANC
E"
 270 LET T1=B
 280 FOR G=0 TO 9
 290 LET F=A(G)/10000
 300 IF F=0 THEN GO TO 440
 310 IF F=1 THEN GO SUB 360
 320 IF F=2 THEN GO SUB 400
 330 PRINT N,A$;B(G),T1
 340 NEXT G
 350 STOP
 351 GO TO 100
 360 LET A$="+"
 370 LET N=A(G)-10000
 380 LET T1=T1+B(G)
 390 RETURN
 400 LET A$="-"
 410 LET N=A(G)-20000
 420 LET T1=T1-B(G)
 430 RETURN
 440 INPUT B$
 450 IF B$="STOP" THEN STOP
 460 GO TO 100
```

Syntactic Sum: -15775, 4K

For 8K ROM users: This program will not fit in 1K RAM with 8K ROM. To use it with more than 1K, delete lines 30-60 and change these lines to:

```
  90 FOR K=1 TO 10
 260 PRINT "NR";TAB 8;"AMOUNT";T
AB 19;"BALANCE"
 280 FOR G=1 TO 10
 300 IF F<1 THEN GOTO 440
 310 IF F<2 THEN GOSUB 360
 320 IF F>=2 THEN GOSUB 400
 330 PRINT N;TAB 8;A$;B(G);TAB 1
9;T1
```

Syntactic Sum: 43566, 8K
With the 8K BASIC you can use decimal numbers.

DEAR EDITOR:

Here is a better way to exit my Inverse Cryptology game (Jun. 81). Change and add these lines:

```
 115 LET R=0
 410 PRINT "ENTER LETTER OR LETT
ER=LETTER
 425 LET Y=CODE(U$)
 430 IF R=Y THEN GO TO 465
 465 LET R=0
 647 IF R=0 AND A(Y-38)<64 THEN
LET R=Y
```

You can still guess letter=letter. If you're guessing the first coded letter of the phrase, just enter the correct letter. When you know the phrase you can type the remaining letters in order, leaving out the coded letters and the equal signs. This program codes up to 69 letters (just over 2 lines) with 1K RAM. It no longer requires an equal sign between the guess and the coded letter. New Syntactic Sum: -3941. (P.S. Will CAI's printer do mailing labels?)

Harold A. Lamkin, Mt. Clemens, MI

The CAI printer works with thermal paper only. I've never seen thermal mailing labels; if they exist, then CAI's printer can print them--AZ

I'd appreciate corrections to errors in programs in 30 Programs for the ZX80 (from Melbourne House). For example, line 650 of Blackjack program, p.40, is incomplete. Also, what changes are necessary to enter these programs in the 8K machine (eg. line 120 on p.76, Life, will not go in as shown in book)? In entering Chomp, p.95, on my 8K ROM (1K) I get 4/100 and a blank screen when I try to run it.

Alan Gayle, Brownsville, TX

According to Michael Sierchio at Image Computer Products, line 650 of Blackjack should read:

```
 650 LET N(I+2)=N(I+2)-10
```

The book contains other errors; Image is sending us a list that we will pass on to you.

Line 120 of Life won't go in because the 8K ROM doesn't use an argument (a number in parentheses) after RND. The 8K RND gives a random number >=0 and <1; the 4K ROM sets the seed of its random number generator to the number in parentheses. To get a random number between 1 and 10 on the 8K ROM, use INT (RND*10+1). RND*10 makes the random number range from 0 to <10; INT drops the decimal. Now the numbers range from 0 to 9, so add 1 to get a number between 1 and 10. To get larger random numbers, increase 10 to 100, 1000, or any power of 10 you need to get the proper number of digits to the left of the decimal point.

Chomp just won't fit in an 8K/1K machine. The 8K ROM eats up more RAM for system variables, cutting down your available RAM. Use Bytes Available (Jun.81 p.13) to see how much memory is left after entering your program (don't type NEW first, just the PRINT line). I found Chomp left only 83 bytes on my 8K/1K machine, not enough to run.--AZ

In using my new 8K ROM, I found something unusual about it. If you ask it to PRINT 2**32, it gives 4294967300. But type PRINT 2**32-1, and you get 1288490200! What an error!! This strange result also happens with higher powers of 2 and numbers like 3**34. RUN this program to see some strange answers:

```
  10 LET A=2**32
  20 PRINT A,A-1,A+1,A-A+1,A-A-1
,A-1-A
  30 STOP
```

David Shulman, Peabody, MA

According to my H-P calculator, 2**32=4294967296. The 8K ROM's 9.5 digit floating point arithmetic accounts for the slight error of 4.

But my calculator gives vastly different answers for A-1 and A-1-A than my 8K ROM does. We also found other arithmetic "irregularities" with numbers >2**26. See 8K ROM story p.1 for Sinclair's plans to replace ROMs with this bug. Not all 8K ROMs have this error; check yours out before you write to Sinclair.--AZ

My 8K ROM and 16K RAM arrived recently. I am disappointed in the power supply. It seems to be poorly regulated and overworked. After several hours the voltage swings wildly but is lower than it should be, and data disappears from the screen, reappearing after it idles for a while. It also "sings" after about 6 hours, another sign of poor regulation. It will probably go back to Sinclair.

One disappointment was that TL$ is no longer a standard function. Never fear! There's an easy way to do the same thing. Line 40 of this demonstration program shows a 1-line trick:

```
   10 INPUT A$
   20 IF A$="" THEN GOTO 60
   30 PRINT LEN A$,A$
   40 LET A$=A$(2 TO LEN A$)
   50 GOTO 20
   60 PRINT "THE STRING IS EMPTY"
   70 STOP
```

Martin Irons, Goshen, NY

My 8K ROM also eats some programs, but data doesn't return. Other readers say they have the same problem. John Strang, Burbank, CA, told us that any program with a PAUSE line crashes if you run it and leave it in command mode. It's OK if you type it in and leave it on the screen, or if you continue to use the program. To check this, type:

```
   1 PAUSE 1
```

RUN, BREAK, then LIST.
The machine will crash in 9 min. 6 sec. To prevent such a crash, POKE 16437,255 after a PAUSE statement. See the note on p. 127 of the ZX81 (8K) manual.--AZ

I thought your "Cassette Eavesdropper" (Dec.80) would help me load programs, but I hate to build things. I found that if I put an AM radio tuned at about 1600 near my ZX80 while loading, I could hear when the computer started and stopped getting data because of the AM signal it puts out while functioning. Try it--you'll be surprised at the neat noises it makes.

Warren Schoettlin, San Marino, CA

I wanted to respond to Bob Irwin's call for a Compuserve ZX80 Special Interest Group (Jun.81) but I am not yet a Compuserve user and his address was not listed. I would like to contact him.

I would also like to hear from anyone who has used his/her ZX80 in a timeshare, TP environment. Information on modem attachment, machine code emulator programs, etc. would be particularly appreciated.

Robert B. Keller, 1004 Skyline Dr.
Rochester, MN, 55901

Contact Bob at his home address: 7710 Braemeadow, Houston, TX, 77071.--AZ

These SYNTAX readers would like to hear from others in their areas. If you would like to contact local ZX80/MicroAce users, send us your name and address. We'll publish them when space permits.--AZ

*Michael Meagher, 4532 E. Colorado Ave., Las Vegas, NV, 89104.

*Lee Wilson, P.O. Box 176, Morganfield, KY, 42437.

*David Chase, 1971 Greenway Park, Blackburn-Hamlet, Ottawa, Ontario, Canada, K1B 5A9, 613/837-3932.

BOOK REVIEW

Title: The ZX80 Companion
From: LINSAC, 68 Barker Road,
Middlebrough, UK, TS5 5ES
Price: $19.95

The ZX80 Companion tries to have something for everyone--from how to plug in your computer to ROM entry points. The first 14 pages cover setup, operation, loading problems, and basic computer theory--possibly useful to beginners. The 27-page chapter on BASIC gives a few new insights over the supplied manual, PEEKs and POKEs around the system variables, and shows how to store machine code in a REM statement (without mentioning that it can crash the system). The 10 pages on construction and hardware give some troubleshooting tips but go no deeper than an edge-connector pinout. Six well-documented LINSAC games comprise the last 24-page chapter. Some contain valuable programming techniques, such as timers and reverse video. There's also a Z80 opcode list in decimal and hex.

But if you're a machine language person, the real meat of the book is the 30-page chapter on the ROM. It gives many BASIC addresses, including entry points and calling conventions for math and screen printing routines. The tape routines are massaged, the stack moved down, and the display file explained in good detail. The screen-and-keyboard routine is discussed, moved, patched, and finally called as a USR routine for a moving display! Good stuff here, although it doesn't explain how the video really works.

This 128-page, 8 1/4 x 6 inch soft-cover book with lie-flat plastic binding may be overpriced at $19.95. But it is a valuable reference and about the only one available. All in all, I'm glad I have a copy--but I'd rather pay half the price for just the ROM chapter!

Jim Williams, Calumet City, IL

(Note: Signed reviews reflect the author's opinions. Unsigned reviews represent SYNTAX's editorial opinion. We welcome any hardware, software or book reviews from SYNTAX readers.)

MACHINE LANGUAGE PROGRAMMING — LOGICAL FUNCTIONS

Many readers have called us about differences in 4K and 8K ROMs. You can use machine language (ML) to adapt programs for both computers or to regain lost uses, in particular, the masking function of 4K logical AND and OR.

ML functions A AND B and A OR B, unlike arithmetic, return numbers representing logical values. A and B are OPERANDS. ML recognizes two values, true (=1) and false (=0).

1 AND 1 is 1	1 OR 1 is 1
1 AND 0 is 0	1 OR 0 is 1
0 AND 0 is 0	0 OR 0 is 0

ML code compares bit positions in A and B to see if they are the same or different. AND returns true (or 1) for each bit position if both bits are 1, while OR returns true if at least one bit of the pair is 1. So AND returns false (or 0) if at least one bit is 0, and OR returns false only if both bits are 0.

BASIC converts the binary strings of 1s and 0s into decimal equivalents. Check the following binary results and their decimal equivalents:

```
00100000 = 50 AND 37 = 32

00110010 = 50
00100101 = 37

00110111 = 50 OR 37 = 55
```

BASIC 4K AND tests bits in SON OF BIG CHARACTERS (Feb.81) before SON makes them into big characters. For example, in line 110, Q AND B (=128) returns 128 (10000000 in binary) if Q contains a 1 in the 128 (or 2**7) binary position, 0 otherwise. (Remember the rightmost position is 2**0, so the eighth from the right is 2**7.) In 4K, you can test any bit position in a binary number for a 1 by typing its decimal equivalent AND one of the following decimal B operands:

BIT # (from right)	B (Dec)	B (Bin)
7	128	10000000
6	64	01000000
5	32	00100000
4	16	00010000
3	8	00001000
2	4	00000100
1	2	00000010
0	1	00000001

AND and OR results from the 8K ROM, on the other hand, cannot mask bits. You don't get decimal numbers made from logical bit comparisons; instead you get A or 0. A AND B returns A if B<> (doesn't equal) 0, and 0 if B=0. A OR B returns 1 when B<>0, and A when B=0.

50 AND 37 is 50	50 OR 37 is 1
50 AND 0 is 0	50 OR 0 is 50

AND and OR on 8K do not return bit-by-bit true or false; they depend on A's value and B's logical value (B zero or nonzero).

As a result, SON OF BIG CHARACTERS must be modified for the

ZX81 or 8K ROM, since AND no longer works as a bit tester. You can duplicate 4K AND in your 8K machine by POKEing USR routines into RAM.

Assembled ML routines from the 4K ROM to put AND and OR into an 8K machine appear for use with an ML monitor (see May 81,p.13). A decimal listing enables you to POKE them in by hand. (We assume you will load both AND and OR routines at the same time.) They differ from the original 4K ROM listing in that they return results to the BC register instead of to HL, and we added INIT and INIT2 so AND and OR would fetch operands from the appropriate locations.

Before LOADing these routines, you can prevent the computer from overwriting them by POKEing a new value into RAMTOP, the 8K's way of knowing where the end of RAM is.

```
POKE 16388,224
POKE 16389,67
NEW (NL)
```

RAMTOP points to the first non-existent RAM address; lowering RAMTOP fools the computer into thinking there is no more RAM space above RAMTOP since it fills RAM from the bottom up. The new RAMTOP is now 17376, the first byte you can use for USR routines. You can't, however, SAVE your routines, because the computer doesn't know they're there.

Before you use 4K BASIC AND or OR, you must POKE A and B (your operands) into 8K memory so ML can load them into the appropriate registers. The following should appear before your USR call:

```
POKE 17376,A-256*(INT A/256)
POKE 17377,INT(A/256)
POKE 17378,B-256*(INT A/256)
POKE 17379,INT(A/256)
```

To run AND, use USR (17380); for OR, USR (17394). Begin POKEing the decimal listing at 17380.

4K BASIC AND and OR also use expressions as operands, but some rules change. More later.

DECIMAL LISTING

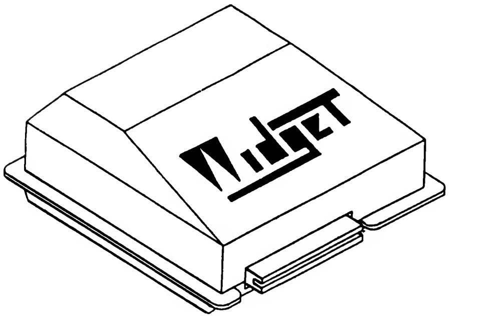
```
 42 224  67 237  91 226  67 124
162  71 125 163  79 201  42 224
 67 237  91 226  67 124 178  71
125 179  79 201
```

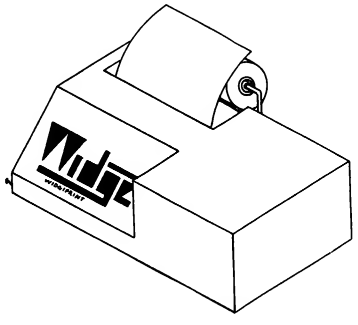
ASSEMBLY LISTING—4K LOGICAL AND & OR FOR 8K MACHINES

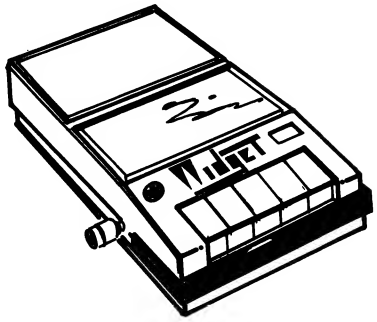
```
43E0           00080        ORG     43E0H    ;ORIGIN PLUS ROOM TO SAVE A AND B
0002           00090        DEFS    2        ;SAVE 2 BYTES FOR A
0002           00095        DEFS    2        ;SAVE 2 BYTES FOR B
43E4 2AE043    00100 INIT   LD      HL,(43E0H)      ;A OPERAND
43E7 ED5BE243  00110        LD      DE,(43E2H)      ;B OPERAND
43EB 7C        00120 AND    LD      A,H
43EC A2        00130        AND     D
43ED 47        00140        LD      B,A
43EE 7D        00150        LD      A,L
43EF A3        00160        AND     E
43F0 4F        00170        LD      C,A
43F1 C9        00180        RET
43F2 2AE043    00190 INIT2  LD      HL,(43E0H)      ;A OPERAND
43F5 ED5BE243  00200        LD      DE,(43E2H)      ;B OPERAND
43F9 7C        00210 OR     LD      A,H
43FA B2        00220        OR      D
43FB 47        00230        LD      B,A
43FC 7D        00240        LD      A,L
43FD B3        00250        OR      E
43FE 4F        00260        LD      C,A
43FF C9        00270        RET
```

THE CAI 'WIDGET'® SERIES OF PERIPHERALS FOR THE ZX-80 OPENS THE DOOR TO REAL COMPUTING

✓ *Check Our Specifications*







WIDGET® $\$49^{95}$

PARALLEL TTL INPUT PORT

- Eight bits non-latched
- Draws one low power Schottky load
- Uses:
  - *Interface to user designed circuits*
  - *Sensing alarm conditions*
  - *Interpreting limit switches*
  - *Accepting user switches or other input devices*

PARALLEL TTL OUTPUT PORT

- Eight bits concurrently latched
- Supplies up to ten (10) low power Schottky loads
- Uses:
  - *Interface to user designed circuits*
  - *Activating Led's, Lamps and Indicators*
  - *Powering relays, SCRS and TRIACS for high current 110 VAC loads*
  - *Sounding alarms*

SERIAL RS-232 INPUT/OUTPUT PORT

- Three wire bi-directional serial line
- Baud rates strappable at 110, 300, 1200, 2400, 9600
- Communicate with:
  - *Serial printers*
  - *Modems (transmit and receive data only)*
  - *Terminals (teletypes and CRT's)*
  - *Graphic plotters*

ADDRESS AND CONTROL SIGNALS from the ZX-80 are all buffered to allow ease of expansion.

WIDGET INTERFACES DIRECTLY TO THE WIDGIPRINT and WIDGITAPE and the 16K RAM and 8K ROM by simply, plugging one into another.

The WIDGET, like all CAI devices, comes with complete schematics and instructions.

WIDGIPRINT® $\$99^{95}$

HIGH QUALITY THERMAL PRINTER at a very affordable price

THE WIDGIPRINT PROVIDES THIRTY-TWO (32) COLUMNS OF OUTPUT in compliance with the format shown on the television screen

EASY INSTALLATION

- The CAI Printer is designed to plug directly into its own port on the Widget board. This leaves all eight I/O bits on the WIDGET board available to the user. There is no soldering or modifications necessary
- The printer is equipped with its own power supply

SPECIFIC PRINTER SPECIFICATIONS

- Full alpha-numeric capabilities with limited graphics
- Thirty-two (32) characters per line
- Self-contained power system
- Two-hundred lines / minute print speed
- Long life — few moving parts
- Paper size 3.15" roll style readily available • Size 6.4" x 1.9" x 3.7"

The PRINTER comes with a complete instruction and operations manual.

WIDGITAPE® $\$99^{95}$

The WIDGITAPE is designed as a ***highly reliable*** mass storage device for the data generated with the ZX-80. This device interfaces through the CAI WIDGET Board, and is completely independent of the tape connections on the ZX-80.

The WIDGITAPE provides much of the functionality found in floppy disks for a fraction of the cost.

THIS INCLUDES

- Writing a file
- Reading a file
- Extending or modifying a file
- Deleting a file
- Each tape also maintains its own file directory which can be viewed on the television screen

All these functions are programmatically accessable to the user via simple keyboard commands.

INSTALLATION

- **The WIDGITAPE plugs into its own plug on the WIDGET Board with no need to solder or modify the ZX-80.**

USE THIS FORM TO ORDER YOUR WIDGET® PERIPHERALS

Send Check, Money Order Or Credit Card Number To:

Quantity	Description	Unit Price	Total
	WIDGET®	$49.95	
	WIDGIPRINT®	$99.95	
	WIDGITAPE®	$99.95	
	TOTAL		

CAI Instruments, Inc.
P.O. Box 446
Midland, MI 48640
517/835-6145

Add $5.00 for shipping and handling

Michigan residents add 4% sales tax

Mastercharge ____________ Visa ____________
Exp. Date ____________
Name ____________
Address ____________
City ____________ State ____________ ZIP ____________

BEGINNERS' DIMENSIONED ARRAYS

Although BASIC uses simple English commands, the DIM statement is one of those that takes a little explanation. We'll look at numerical arrays this month, using single letter names and containing numbers. Later we'll consider string arrays, which use letters followed by a $ (A$, for example) and contain letter expressions.

DIM stands for DIMensioned array. Think of an array as a matrix or a table. Accountants may know them as spreadsheets. "Dimensioned" means that you can specify its size. The 4K ROM takes single dimensioned arrays, meaning that you get 1 row but many columns in your table. The 8K ROM permits multidimensioned arrays, so you can have several rows and columns.

Arrays have single letter names, A-Z. Specify the size of the array in parentheses after its name. DIM A(3) means set up an array called A with 4 elements in the 4K ROM. The first element of A is labelled A(0), the second A(1), the third A(2), and the fourth A(3). Each array A(n) contains n+1 elements. Here's the table for DIM A(3) in the 4K ROM:

A(0)	A(1)	A(2)	A(3)	A(4)

The 8K ROM counts elements differently. DIM A(3) means set up an array with 3 elements. The first is A(1), the second A(2), and the third A(3). There is no A(0). So for 8K ROMs, any array A(n) has n elements.

DIM A(2,3) sets up a 2-dimension array A; think of it as having 2 rows and 3 columns. The first number in parentheses tells what row, the second what column. Each element requires a row and column number to be identified. Here's the table for DIM A(2,3):

A(1,1)	A(1,2)	A(1,3)
A(2,1)	A(2,2)	A(2,3)

What do you use arrays for? They store data. You can set up an array at the beginning of your program and then input values into it using a FOR-NEXT loop. You can use your array elements in arithmetic calculations or PRINT statements. Try this sample program on your 4K ROM:

```
  10 DIM Z(3)
  20 FOR I=0 TO 3
  30 PRINT "Z(";I;")=";
  40 INPUT Z(I)
  50 PRINT Z(I)
  60 NEXT I
  70 PRINT Z(0)+Z(1),Z(1)*Z(2),Z
(3)/Z(0),Z(1)-Z(2)
```

Line 10 sets aside room in memory for your array. Lines 20-40 allow you to enter the numbers you want in your array. Line 30 tells you which element you are entering; line 50 prints the value you typed. Line 70 shows how you can perform arithmetic functions on and print array elements. (This program also works on 8K ROMs if you change line 20 to FOR I=1 TO 3 since the 8K ROM doesn't allow 0 as a subscript for arrays and change Z(0) in line 70.)

If you have an 8K ROM, try this sample program to see a multidimensioned array:

```
  10 DIM Z(3,3)
  20 FOR I=1 TO 3
  30 FOR J=1 TO 3
  40 PRINT "Z(";I;",";J;")=";
  50 INPUT Z(I,J)
  60 PRINT Z(I,J)
  70 NEXT J
  80 NEXT I
```

Each line's function is similar to the comparable line in the 4K ROM program. We nested (put 1 inside the other) 2 FOR-NEXT loops to enter values for I and J. Notice that each element Z(I,J) contains 1 number (of any number of digits) although it is named by 2 coordinates I and J. The value of the element has no relation to its coordinates. Add a line like 70 above to check that you can perform arithmetic on and print multidimensioned array elements as well. Remember to name each element by specifying an I (row number) and a J (column number).

BEGINNERS--ANALYZING THE PROBLEM

The first step in solving a problem with a computer is analyzing how you would solve it without one. Both the human brain and the computer can accomplish these 4 tasks: input, output, storage, and processing.

If you ask a person to add 2 numbers, he must: 1) hear what you ask him and transmit your question to his brain, 2) remember the numbers he must add, 3) add the numbers, and finally 4) tell you the sum.

These steps are equivalent to 1) input, 2) storage, 3) processing, and 4) output. A computer follows the same pattern to solve a problem. To add any 2 numbers using a BASIC program, follow the same steps, using a special vocabulary that your computer can understand, the BASIC language. Here's a program to add any 2 numbers (within the machine's range) and print the result:

```
  10 INPUT A
  20 INPUT B
  30 LET C=A+B
  40 PRINT C
  50 STOP
```

The INPUTs in lines 10 and 20 tell the computer to wait for the person at the keyboard to type 2 numbers and then store those numbers as variables A and B. These lines correspond to steps 1 and 2 of our problem-solving process, input and storage.

Line 30 tells the ZX80 to add the value of A to the value of B and call the answer C. This is step 3, processing, and back to step 2 (because the computer also must store the results of processing).

Line 40 prints C, the sum. This is step 4, output. After all, what good is it if the computer knows the answer and you don't? Line 50 tells the computer that it's done (usually unnecessary in such a simple program).

To this skeleton program you can add some frills, like PRINT statements to prompt the user to input A and B:

```
   5 PRINT "ENTER FIRST NUMBER"
  15 PRINT "ENTER SECOND NUMBER"
```

Well written programs include such prompts if the memory space permits them. Conversely, when you run out of space in a long program, the first things to ax are PRINT statements just there to make the program convenient.

Whenever you want to solve a problem on your computer, your first job is to break the problem down into steps that your computer can do, starting with requesting and storing necessary data. Then tell it what to do with that data, such as perform arithmetic on it (add, multiply, etc.). Finally you instruct it to tell you the results, usually with a PRINT statement.

Richard Forsen, New Hartford, NY

CLASSIFIED ADS

SYNTAX CLASSIFIEDS: Just $2.75/line (4 line min.).. Send ad, typed 35 char/line, with payment by 15th of any month for next month's issue. Classified Ads, SYNTAX, RD 2 Box 457, Bolton Rd, Harvard, MA, 01451.

ZX80/81 "Record"--at last a tape record system! Save, load or enter new 96 byte records. Ideal for addresses, etc. Listing for all 1K machines (4K/8K ROM) $9. "Directory"--a simple program to read tapes and display program names. Listing (8K ROM) $6. Logan Software, 24 Nurses Ln, Skellingthorpe, Lincoln, LN6 0TT, UK

BASIC programs: Symbolic algebra, video utilities, poetry, monitor, more. Send SASE for info: R.Holmes 1303 E.Wilson #204,Batavia IL 60510

For Sale: MICROACE ZX-80 (4K ROM & 2K RAM), subscriptions to SYNTAX & Sync + all back issues + Sinclair Manual. Like new, $110 516-921-3603

Zilog Z80-Z80A CPU Technical Manual, $7.50, & Z80-Assembly Language Programming Manual, $15. Add 5% for postage and handling. Send check/credit card no. to SYNTAX, RD 2 Box 457, Bolton Rd., Harvard, MA, 01451.

Video secrets revealed! This 10-page document explains in detail how ZX80 video works, including a USR routine to run the display. Also included are an explanation and USR routine to run the keyboard and some notes on ROM routines. $4 postpaid. Jim Williams, 262 Chappel, Calumet City, IL, 60409.

FOR SALE: Used ZX80 with first 12 SYNTAX issues. $100/offer. Paul Cosby, RD 2 Box 117 Canary Rd, Quakerstown, Penn. 18951.

MAKE MUSIC with your 4K-1K machine 2 octave range, 127 note length, any tempo endlessly repeated or listen to random notes. cassette $6.95. $10 outside USA. Postpaid. William Don Maples 688 Moore St. Lakewood, Colorado 80215.

4K/1K PROGRAM LISTINGS..F1-Wisard's Lair I/II..F2-High Low Number..F3-Fun with Graphics..E1-Anti-Aircraft (8K/1K)..F1&F2&E1 2K RAM into..$1 or 3 for $2.50...M. E. Meagher,4532 Colorado Ave., Las Vegas, NV 89104.

Keyboard beepers-simple three wire install $10 ppd. Keyed 46 pin buss connectors, $5 ppd. Gene A Langley 4216 Galaxy Dr Union City CA 94587

SYNTAX

Next month:
INPUT PORT

THE
HARVARD
GROUP
Bolton Road, Harvard, Mass. 01451

First Class